CISCO router malware

Detection Strategy

Version 2.0 – 28/04/2016,

LE JAMTEL Emilien, CERT-EU

TLP-AMBER

Table of contents

1 Introduction

Based on knowledge we retrieved from a specific CISCO router malware, we were able to elaborate a detection strategy for several actions performed by the Threat Actor.

The first part details detection strategies that can be used if you have the capabilities to put inline-sensor to your network or have PCAP of network traffic.

The second part details detection strategies if you only have connection logs (netflow for example) and are not able to perform deep-packet inspection.

Last part is a rule that can be used to locally detect presence of the backdoor.

2 Detection strategy with full-packet capture capabilities

2.1 Detection of ICMP packets used to control the backdoor

The forged ICMP request used to control the backdoor have some particularities:

- itype : 8 (echo request)
- ICMP ID : multiple of 4 (in binary mode, must end with 00)
- ICMP sequence number : 2
- icode : 0 (standard)
- ttl : over 200
- size of payload : Can be non-standard but mostly depend of the action performed on compromised devices. However in the samples analyzed, the size of the payload was 44 bytes.

SNORT RULES

```
## Rule to detect the typical ICMP packets without icmp_id parameter

alert icmp any any -> any any  (msg:"CERT-EU - CISCO ICMP backdoor"; itype: 8; icmp_seq: 2; ttl:>200; sid:1811914; rev:3; classtype:bad-unknown;)


## Rule to detect the typical ICMP packets with icmp_id=8, as observed in the analyzed samples

alert icmp any any -> any any  (msg:"CERT-EU - CISCO ICMP backdoor - ID8"; itype: 8; icmp_seq: 2; ttl:>200; icmp_id:8; sid:1811915; rev:3; classtype:bad-unknown;)


## Rule to detect the typical ICMP packets with icmp_id=8, as observed in the analyzed samples, with a payload of 44 bytes

alert icmp any any -> any any  (msg:"CERT-EU - CISCO ICMP backdoor - ID8 - dsize44"; itype: 8; icmp_seq: 2; ttl:>200; icmp_id:8; dsize:44; sid:1811916; rev:3; classtype:bad-unknown;)
```

It was not possible to write a Snort rule that check if the icmp_id is a multiple of 4. However in the analysis, we discovered that the threat actor set this value to 8 (see 2nd SNORT rule).

We know that the payload can be of any size, but in the, payload was always 44 bytes. However it seems that the Threat Actor was always performing the same actions, so other actions on the compromised devices could be using longer or shorter payload.

SOURCEFIRE RULE

To solve the issue with icmp_id being a multiple of 4, Sourcefire developed a rule to detect these packets.

This rule is available for all Sourcefire users but must be manually activated.

```
## Rule number: 38330
```

SURICATA

With Suricata it is possible to get the raw ICMP data, using LUA signatures. Of course Suricata will need to be compiled with luajit support..

```
## Rule

alert icmp any any -> any any (msg:"CERT-EU - ICMP backdoor - icmp_id match";
itype: 8; icmp_seq: 2; ttl:>200; luajit:test.lua; classtype:trojan-activity;
sid:1; rev:1;)


## Content of the test.lua file
function init (args)
    local needs = {}
    needs["packet"] = tostring(true)
    return needs
end

function match(args)
      local ip_offset = 14 -- Assuming it's ETHER packet
      local icmp_id_offset = 39

      v = args["packet"]
      b1, b2 = string.byte(v, icmp_id_offset), string.byte(v, icmp_id_offset+1)
      icmp_id = (b2 + b1 * 256)
      if (icmp_id % 4) == 0 then
      # match
            return 1
      end
      # no match
      return 0
end

return 0
```

Course of Action

If those rules trigger an alert and the target of the ICMP packet is a CISCO device, it is possible the targeted device is actually compromised.

All packets coming from the source IP should be deeply reviewed and the network device should be analyzed.

2.2 Detection of typical NMAP scans

Threat Actor is looking for CISCO network devices on the Internet. To do that, they perform NMAP scans for specific ports (and only those ports):

- 21 (ftp)
- 22 (ssh)
- 23 (telnet)
- 80 (http)
- 443 (https)
- 2001 (cisco telnet)

More specifically, they use the following call in a .NET application:

```
   Program.proc_start("nmap", "-sS -PN -n -O -p21,22,23,80,443,2001 -oN " + str
+ ".ip0 " + str);
```

Options for nmap are:

- `sS`: TCP SYN scan
- `PN`: skip Ping Host discovery
- `N`: no DNS resolution
- `O`: OS detection

A SNORT rule based on these details will probably generate a lot of false positives because we cannot specify the fact that only those ports were targeted by the scan.

Most IDS already have TCP SYN scan detection. This functionality should be exploited and filtered to identify IP scanning only for those ports.

OS detection option can also generate traffic outside of these defined TCP ports (TCP packet sent to high number port)

If any IP is identified, it should be considered as potential Command and Control server.

2.3 Detection of CISCO configuration extract

Threat actor is using FTP or TFTP protocols to extract running configuration from the CISCO network devices.

The filenames used when extracting the data follow 4 patterns :

- IPaddress1_IPaddress2.txt
- IPaddress1_IPaddress2.bin1
- SNMPString_IPaddress1_IPaddress2.txt
- SNMPString_IPaddress1_IPaddress2.bin1

We also identified some keywords used in FTP commands (usernames, passwords, filenames).

Actually, any connection from a network device to a FTP/TFTP server outside of the legitimate perimeter is highly suspicious...

SNORT RULES

```
## rule for TFTP extraction - based on typical filename used by the threat actor
alert udp any any <> any 69 (msg: "CERT-EU - CISCO TFTP extraction"; pcre:"/([0-9]{1,3}\.){3}([0-9]{1,3})_([0-9]{1,3}\.){3}([0-9]{1,3})\.(bin1|txt)/" ;sid:1811922; rev:1; classtype:bad-unknown;)

## rule for FTP - based on filename + keyword
alert tcp any any <> any 21 (msg: "CERT-EU - CISCO FTP extraction"; pcre:"/([0-9]{1,3}\.){3}([0-9]{1,3})_([0-9]{1,3}\.){3}([0-9]{1,3})\.(bin1|txt)|largo|Pedro|timeout|ccrthwtd|rw4orion|cisco123|ndf/" ;sid:1811923; rev:1; classtype:bad-unknown;)

## general rule for outbound TFTP Data Transfer with Cisco Config - not specific to our current case
alert udp $HOME_NET any -> $EXTERNAL_NET 69 (msg:"ET TFTP Outbound TFTP Data Transfer with Cisco config"; content:"|00 03|"; depth:2; content:"|0a 21 0a|version|20|"; distance:2; within:12; classtype:policy-violation; sid:2015857; rev:4;)
```

The CISCO device is always initiating the FTP/TFTP connection

Course of Action

Those rules detect upload of running configuration to a FTP or TFTP server based on behavior from the threat actor.

If any of the first two rules is triggered, the IP hosting the FTP/TFTP server is a CnC server but there is no way to say for sure that the CISCO device is compromised. Further analysis of data exchange between the Command and Control server and the device should be deeply reviewed and the CISCO device should be analyzed

2.4 Detection of SNMP requests

Threat actor is using snmpset to request extraction of running configuration to a FTP or TFTP server.

```
## rules for SNMP traffic - OID

## FTP extraction

alert udp any any -> any 161 (msg:"CERT-EU - CISCO SNMP OID export FTP"; content: "|2B 06 01 04 01 09 09 60 01 01 01 01 05 85|"; sid:1811919; rev:2; classtype:bad-unknown;)

## TFTP extraction

alert udp any any -> any 161 (msg:"CERT-EU - CISCO SNMP OID export TFTP"; content: "|2B 06 01 04 01 09 02 01|"; sid:1811920; rev:2; classtype:bad-unknown;)


## FTP/TFTP extraction (replace the 2 previous rules but more resource-consuming)

alert udp any any -> any 161 (msg:"CERT-EU - CISCO SNMP OID export FTP AND TFTP"; content: "|2B 06 01 04 01 09|"; pcre:"/\x2B\x06\x01\x04\x01\x09\x02\x01|\x09\x60\x01\x01\x01\x01\x05\x85/s" ; sid:1811921; rev:2; classtype:bad-unknown;)
```

Course of Action

Those rules detect attempts of running configuration extraction. Several actors could be trying to do this kind of actions using the same method and any successful attempt to do that should be considered as a major incident.

The only specific rule is the first one (keywords). If this one trigger an alert, the source IP can be considered as a Command and Control server but there is no way to say for sure that the CISCO device is compromised. Further analysis of data exchange between the Command and Control server and the device should be deeply reviewed and the CISCO device should be analyzed

3 Detection Strategy with connection logs only

3.1 Suspicious ICMP/SNMP combination

In two script discovered, the Threat Actor were using one ICMP packet to modify configuration of the infected device and then SNMP to request configuration extraction via FTP/TFTP.

Between those 2 commands, the scripts use "sleep 5".

Recommended course of action to detect such behavior (and potentially infected devices):

```
-  look for ICMP echo request packet addressed to network devices (example:
   192.168.1.1:0 -> 192.168.1.2:8.0)
-  look for SNMP packet 5/10 seconds later (same IP source of course)
```

3.2 Suspicious SNMP/FTP|TFTP combination

Threat Actor use SNMP to request download of the running config via FTP or TFTP. Via FTP, 8 SNMP request are needed. For TFTP, only 1 SNMP request is needed.

In both case, if successful, the compromised device will initiate the connection

Look for following pattern:

```
## for TFTP
 Malicious_IP:random_port -> Device IP:161
 Malicious_IP:69 <- Device IP:random_port


## for FTP (expecting 8 SNMP request but 4 is enough for detection)
 Malicious_IP:random_port -> Device IP:161
 Malicious_IP:random_port -> Device IP:161
 Malicious_IP:random_port -> Device IP:161
 Malicious_IP:random_port -> Device IP:161
 Malicious_IP:21 <- Device IP:random_port
```

3.3 Huge amount of SNMP packets

We observed that infected devices receive a lot of SNMP request from the Command and Control server on a short period of time.

As we know, netflow can be partial. So any multiplication of SNMP request coming from a unique IP address should be investigated.

3.4 Suspicious traffic from CISCO devices

A functionality of the backdoor is to mirror part of the traffic to an exfiltration address. In some cases the IP source of this traffic is one of the IP of the compromised device.

A large amount of traffic with a CISCO device as IP source is strongly suspicious and should be investigated.

4 Device analysis

If you suspect a device has been compromised, there are several checks to perform. However, as we highly suspect the threat actor is able to capture commands used by the network administrators, it is recommended to use this possibility after some time of network capture.

4.1 Updated ROMMON

The easiest and most convenient way to detect if the ROMMON has been upgraded is to execute the command show rom-monitor from the command line interface (CLI).

```
Router#show rom-monitor
ReadOnly ROMMON version:
System Bootstrap, Version 12.4(1r) [hqluong 1r], RELEASE
SOFTWARE (fc1)
Copyright (c) 2005 by cisco Systems, Inc.
Upgrade ROMMON version:
System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2006 by cisco Systems, Inc.
Currently running ROMMON from Upgrade region
ROMMON from Upgrade region is selected for next boot
```

4.2 Syslog events

You can detect extraction of running configuration via SNMP in Syslog events.

The log will give you the IP address of the exfiltration server (FTP/TFTP)

Here is an example of such entry in the logs:

```
Sep 10 08:04:43.523: %SYS-4-SNMP_WRITENET: SNMP WriteNet request.
Writing current configuration to xxx.xxx.xxx.xxx
Sep 10 08:04:44.523: %SYS-4-SNMP_WRITENET: SNMP WriteNet request.
Writing current configuration to xxx.xxx.xxx.xxx
```

4.3 Unexpected crash/reboot

After deploying the modified rommon in the upgrade region, the device needs to be rebooted. We also observed that some actions performed by the Threat Actor can lead to a crash of the network device.

Any unexpected reboot or crash of potentially targeted devices should be investigated.

4.4 YARA rule

The following Yara rule can be used to detect the presence of the backdoor in a compromised Cisco device.

This should be run against a coredump of the device (`write core` command).

YARA RULE

```
rule ios_patch_sig
{
  strings:
      $ios_patch_sig = {3C 19 ?? ?? 27 39 ?? ?? 03 20 C0 09 [0-40] AF A0 00 20}
  condition:
      $ios_patch_sig
}
```